In the Claims

Please amend the claims as follows:

C1

1. (Previously Amended) The structure of a subpipelined translation embodiment providing binary compatibility between a base architecture and migrant architecture of a VLIW architecture comprising:

a VLIW architecture comprising a base architecture and a migrant architecture and having a base execution mode and a migrant execution mode;

an instruction fetch unit for simultaneously fetching from memory a group of a plurality of instructions, each such group forming a fetch packet, ~~the~~ said instruction fetch unit assigning each fetch packet an operating mode in dependence upon the execution mode at the time the request was made to the memory for the fetch packet;

a shared datapath by both the base and migrant architectures for parsing said base architecture mode and migrant architecture mode fetch packets into execute packets of instructions within said fetch packet that can be executed simultaneously;

a base architecture control circuit for dispatching execute packet instructions having a base execution mode;

a migrant architecture control circuit for dispatching execute packet instructions having a migrant execution mode;

a base architecture decode connected to said shared datapath and said base architecture control circuit for decoding an execute packet in said base mode and generating a corresponding machine word;

a migrant architecture decode connected to said shared datapath and said migrant architecture control circuit for decoding an execute packet in said migrant mode and generating a corresponding machine word;

a multiplexer having at least two inputs and one machine word output wherein one input is the machine word output of said migrant architecture decode and the other input is the machine word output of said base architecture decode, said multiplexer choosing in dependence upon the operating mode of said fetch packet; and

execute hardware connected to said multiplexer for executing execute packet instructions on execution units corresponding to said machine word chosen by said multiplexer.

2. (Previously Amended) The structure according to Claim 1, and further comprising a third input to said multiplexer wherein said third input is a no operation instruction machine word.

3. (Original) The structure according to Claim 1, wherein said machine word also controls registers.

4. (Original) The structure according to Claim 1, wherein said machine word controls a global register file, which supplies operands to all hardware execution units and accepts results of all hardware execution units.

5. (Original) The structure according to Claim 4, wherein said machine word controls local register files that supply operands to either local execution hardware functional units or neighbor hardware execution functional units subsequent to said machine word controlling said global register file.

6. (Original) The structure according Claim 5, wherein said machine word controls the various types of execution hardware that evaluate functions on the operands to produce the results of said hardware execution units subsequent to said machine word controlling said local register files.

C1

7. (Original) The structure according to Claim 1, wherein the base and migrant architecture decode units translates opcodes to the control signals required to execute the specified instructions on the execution hardware functional units.

8. (Currently Amended) The structure according to claim 1, and further comprising said migrant architecture control circuit for issuing no-operation instruction to preserve the semantics of the instruction in the migrant architecture due to differences in instruction latencies between the base architecture and the migrant architecture.

9. (Original) The structure according to Claim 1, wherein said VLIW architecture is a Digital signal Processor (DSP).

10. (Previously Amended) A method of providing binary compatibility between a base architecture and migrant architecture of a VLIW architecture comprising the steps of:

simultaneously fetching from a memory a group of a plurality of instructions, each such group forming a fetch packet~~,~~;

assigning each fetch packet ~~having~~ an operating mode in dependence upon the execution mode at the time the request was made to the memory for the fetch packet;

parsing said base architecture mode and migrant architecture mode fetch packets into execute packets of instructions within said fetch packet that can be executed simultaneously;

dispatching execute packet instructions having a base execution mode;

dispatching execute packet instructions having a migrant execution mode;

C1

decoding an execute packet in said base mode and generating a corresponding machine word;

decoding an execute packet in said migrant mode and generating a corresponding machine word;

choosing one machine word output, in dependence upon the operating mode of said fetch packet, between the machine word decoded in said migrant mode and the machine word decoded in said base mode;

controlling the execution hardware units with said chosen machine word.

11. (Currently Amended) The method according to Claim 10, and further comprising choosing between the output of said migrant architecture decode and the output of said base architecture decode ~~input~~ and a no operation machine word.

12. (Original) The method according to Claim 10, and further comprising controlling registers with said machine word.

13. (Original) The method according to Claim 10, and further comprising controlling a global register file with said machine word, which supplies operands to all hardware execution units and accepts results of all hardware execution units.

14. (Original) The structure according to Claim 13, and further comprising controlling local register files that supply operands to either local execution hardware functional units or neighbor hardware execution functional units subsequent to said controlling said global register file.

15. (Original) The method according Claim 14, and further comprising controlling the various types of execution hardware that

evaluate functions on the operands to produce the results of said hardware execution units subsequent to controlling said local register files.

16. (Original) The method according to Claim 10, and further comprising translating opcodes to the control signals required to execute the specified instructions on the execution hardware functional units within the base and migrant architecture decode units.

17. (Original) The method according to claim 10, wherein said VLIW architecture is a Digital Signal Processor (DSP).

18. (Original) The method according to Claim 10 and further comprising the step of issuing no-operation instruction from said migrant architecture control circuit, to preserve the semantics of the instructions in the migrant architecture due to differences in instruction latencies between the base architecture and the migrant architecture.